

Equipo

2

Trabajo de Programación Avanzada

Manuel Benítez Sánchez

Francisco López Baena

Edwin Mauricio Quishpe Maldonado



ÍNDICE

1) Descripción del problema	pág. 2
2) Objetivos.....	pág. 3
3) Requisitos	pág. 4
4) Diagrama secuencias	pág. 5
5) Diagrama de clases	pág. 6
6) Esquema funcional	pág. 7
7) Diseño BBDD	pág13
8) Manual de usuario.....	pág14
9) Manual de administrador	pág18
 Notas	 pág. 19

1) Descripción del problema

Actualmente los campus universitarios tienen superficies y dimensiones muy extensas, además de una gran cantidad de lugares de interés tanto para estudiantes como para profesores o cualquier visitante, y en muchas ocasiones tardamos demasiado tiempo en conocer el sitio o aprender a movernos sin problemas por él. Y esta dificultad es aún mayor cuando se trata de estudiantes, profesores o visitantes que sólo van a estar durante un breve periodo de tiempo, pues, para cuando son capaces de orientarse dentro del campus su estancia está a punto de finalizar.

Por todo esto pensamos que sería de gran utilidad y de mucha ayuda, desarrollar una herramienta que sirviera no sólo para encontrar aulas, edificios, cafeterías, despachos, instalaciones deportivas, paradas de autobús... etc. sino también para encontrar nuestra propia situación dentro del campus. E incluso, para descubrir lugares de interés dentro del mismo, así como para obtener toda la información que necesitas para llegar a él.

Y a partir de esta lúcida idea, hemos desarrollado una aplicación con la finalidad de que el usuario de la misma tenga al alcance de su mano el medio para poder localizar cualquier punto o lugar de interés que esté dentro del campus.

Son muchas las posibles variables que debíamos abarcar en nuestro proyecto, pues no se limita a un simple callejero del Campus de la Universidad Pablo Olavide, si no que hemos querido ir más allá. De este modo, hemos querido gestionar asignaturas, grados, profesores, aulas, edificios, despachos, y todos y cada uno de los lugares que considerados relevantes para el día a día en nuestra Universidad (cafeterías, quioscos, paradas de bus, metro, zonas de deportes...). Hemos incluido, además de la gestión de estas entidades, mapas de ubicación de las mismas.

La aplicación desarrollada posee una interfaz con la que directamente se accede a los distintos lugares que puedes visualizar en los mapas, como edificios, cafeterías, aparcamientos... Tiene también la opción de ingresar en el sistema, de este modo, una vez logueado podrás gestionar: asignaturas, profesores, aulas, edificios, despachos, grados y lugares, teniendo la posibilidad de crear, eliminar, consultar o modificar cualquiera de ellos.

La herramienta está dirigida especialmente al personal del ámbito académico, es decir, profesores o alumnos, pues no hemos considerado relevante tratar de lugares secundarios como puede ser el de limpieza, mantenimiento... o despachos de autoridades encargadas de la gestión de la universidad en un ámbito no educativo.

Esperamos que nuestra aplicación sea de gran utilidad para todo aquél que en alguna ocasión se ha sentido desorientado o perdido en nuestra enorme Universidad, y que consigamos hacer el día a día mucho más fácil.

2) Objetivos

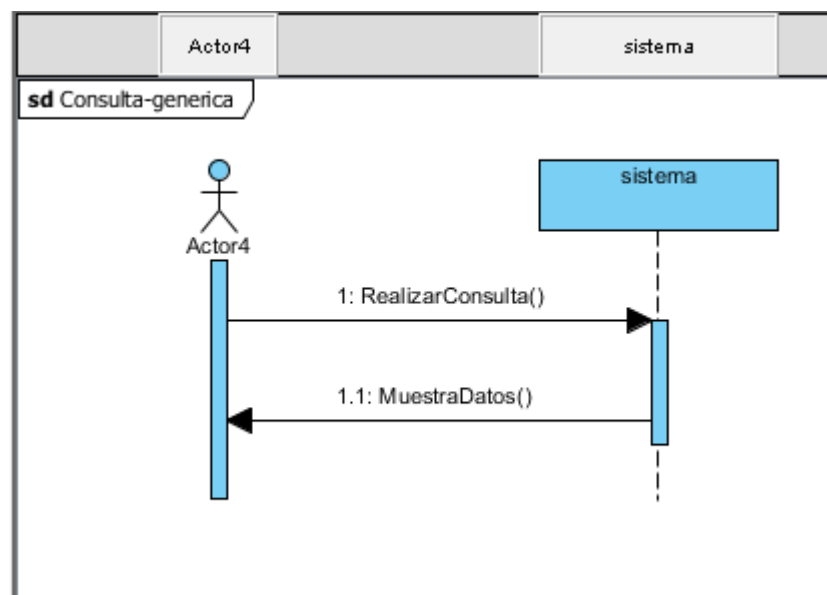
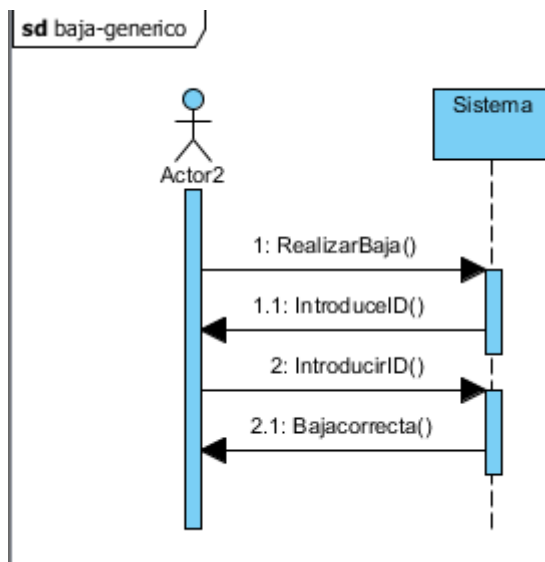
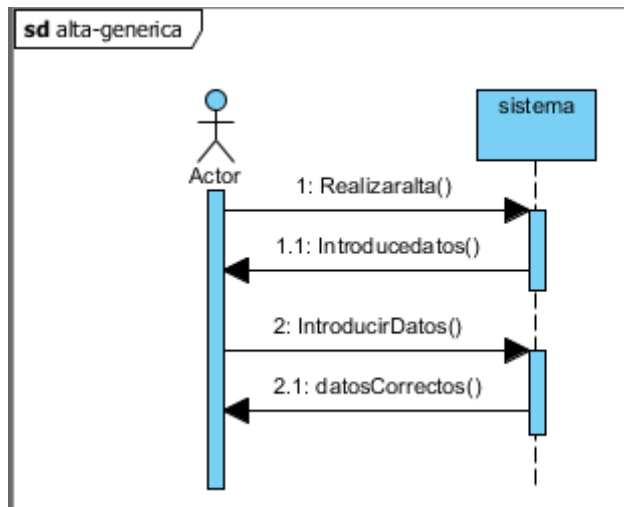
- Debe permitir dar de alta asignaturas
- Debe permitir dar de baja asignaturas
- Debe permitir consultar datos de asignaturas
- Debe permitir modificar los datos de las asignaturas
- Debe permitir dar de alta profesores
- Debe permitir dar de baja profesores
- Debe permitir consultar datos de profesores
- Debe permitir modificar los datos de los profesores
- Debe permitir dar de alta grados
- Debe permitir dar de baja grados
- Debe permitir consultar datos de grados
- Debe permitir modificar los datos de los grados
- Debe permitir dar de alta aulas
- Debe permitir dar de baja aulas
- Debe permitir consultar datos de aulas
- Debe permitir modificar los datos de los aulas
- Debe permitir dar de alta edificios
- Debe permitir dar de baja edificios
- Debe permitir consultar datos de edificios
- Debe permitir modificar los datos de los edificios
- Debe permitir dar de alta lugares de interés
- Debe permitir dar de baja lugares de interés
- Debe permitir consultar datos de lugares de interés
- Debe permitir modificar los datos de los lugares de interés
- Debe permitir dar de alta despachos
- Debe permitir dar de baja despachos
- Debe permitir consultar datos de despachos
- Debe permitir modificar los datos de despachos
- Debe permitir visualizar en el mapa un edificio
- Debe permitir visualizar en el mapa un despacho
- Debe permitir visualizar en el mapa cafeterías, restaurantes.
- Debe permitir visualizar en el mapa aparcamientos
- Debe permitir visualizar en el mapa instalaciones deportivas
- Debe permitir visualizar en el mapa lugares de gestión
- Debe permitir restringir la gestión del sistemas a solo usuarios con permisos
- Debe permitir el envío de sugerencias por email.

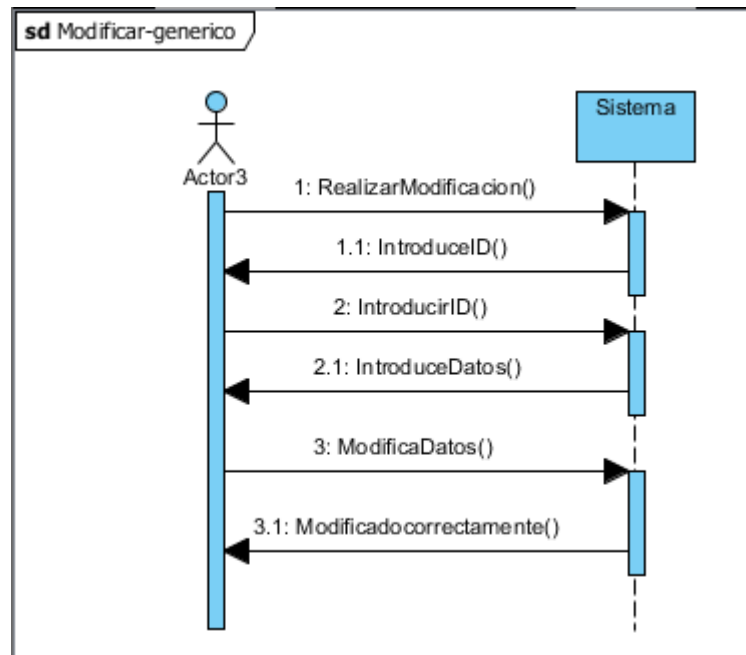
3) Requisitos

1. El sistema debe permitir el alta de nuevas asignaturas introduciendo nombre, y el grado al que pertenece
2. El sistema debe permitir la baja a una asignatura existente.
3. El sistema debe permitir la consulta de los datos de asignaturas
4. El sistema debe permitir la modificación de los datos de una asignatura existente.
5. El sistema debe permitir el alta de nuevos profesores introduciendo nombre, apellidos, id de la asignatura que impartirá y la id de su despacho.
6. El sistema debe permitir la baja a un profesor existente.
7. El sistema debe permitir la consulta de los datos de profesores
8. El sistema debe permitir la modificación de los datos de un profesor existente.
9. El sistema debe permitir el alta de nuevos grados introduciendo nombre.
10. El sistema debe permitir la baja a un grado existente.
11. El sistema debe permitir la consulta de los grados
12. El sistema debe permitir el alta de nuevas aulas introduciendo el número del edificio, la planta donde se encuentra, el id de la asignatura que se impartirá en ella y un comentario.
13. El sistema debe permitir la baja a un aula existente.
14. El sistema debe permitir la consulta de los datos de aulas
15. El sistema debe permitir la modificación de los datos de un aula existente.
16. El sistema debe permitir el alta de nuevos edificios introduciendo nombre, la ubicación, el número de plantas y un comentario del mismo.
17. El sistema debe permitir la baja a un edificio existente.
18. El sistema debe permitir la consulta de los datos de edificios
19. El sistema debe permitir la modificación de los datos de un edificio existente.
20. El sistema debe permitir el alta de nuevos lugares introduciendo su ubicación, el tipo y una descripción del mismo.
21. El sistema debe permitir la baja a un lugar existente.
22. El sistema debe permitir la consulta de los lugares existentes.
23. El sistema debe permitir la modificación de los datos de un lugar existente.
24. El sistema debe permitir el alta de nuevos despachos introduciendo el número del edificio, la planta y el número del despacho.
25. El sistema debe permitir la baja a un despacho existente.
26. El sistema debe permitir la consulta de los despachos existentes.
27. El sistema debe permitir la modificación de los datos de un despacho existente.
28. El sistema debe permitir visualizar los edificios existentes en un mapa seleccionando ver edificios
29. El sistema debe permitir visualizar en el mapa cafeterías, restaurantes...
Seleccionando ver comidas.
30. El sistema debe permitir visualizar en el mapa aparcamientos. Seleccionando ver aparcamientos.
31. El sistema debe permitir visualizar en el mapa medios de transportes (bus, metro...).
Seleccionando ver transporte.
32. El sistema debe permitir visualizar en el mapa instalaciones deportivas (pista de futbol, baloncesto...). Seleccionando ver instalaciones deportivas.
33. El sistema debe permitir visualizar en el mapa de puntos de gestión de la universidad (registro general, casa...). Seleccionando ver para estudiantes.
34. El sistema debe permitir restringir el área de gestión a los usuarios con permisos.
35. El sistema debe permitir enviar email con sugerencias sobre la aplicación.

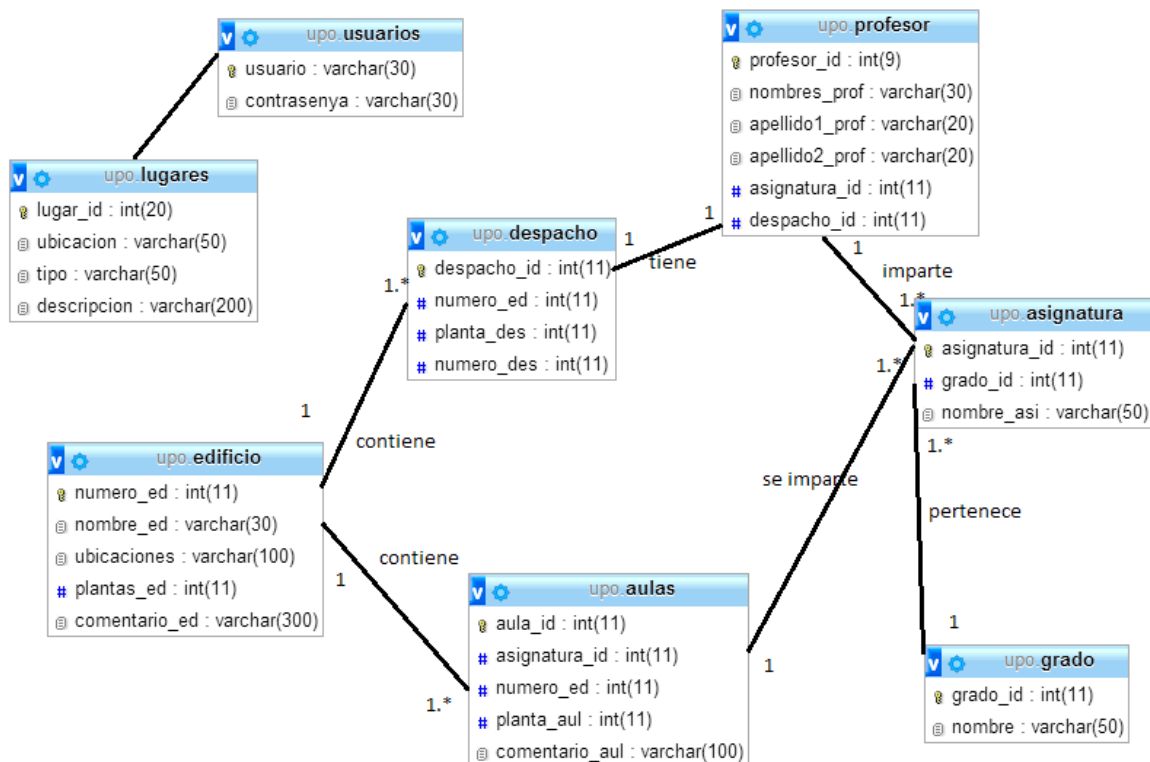
4) Diagrama de secuencias

Los diagramas de secuencias son genéricos para todas las entidades.



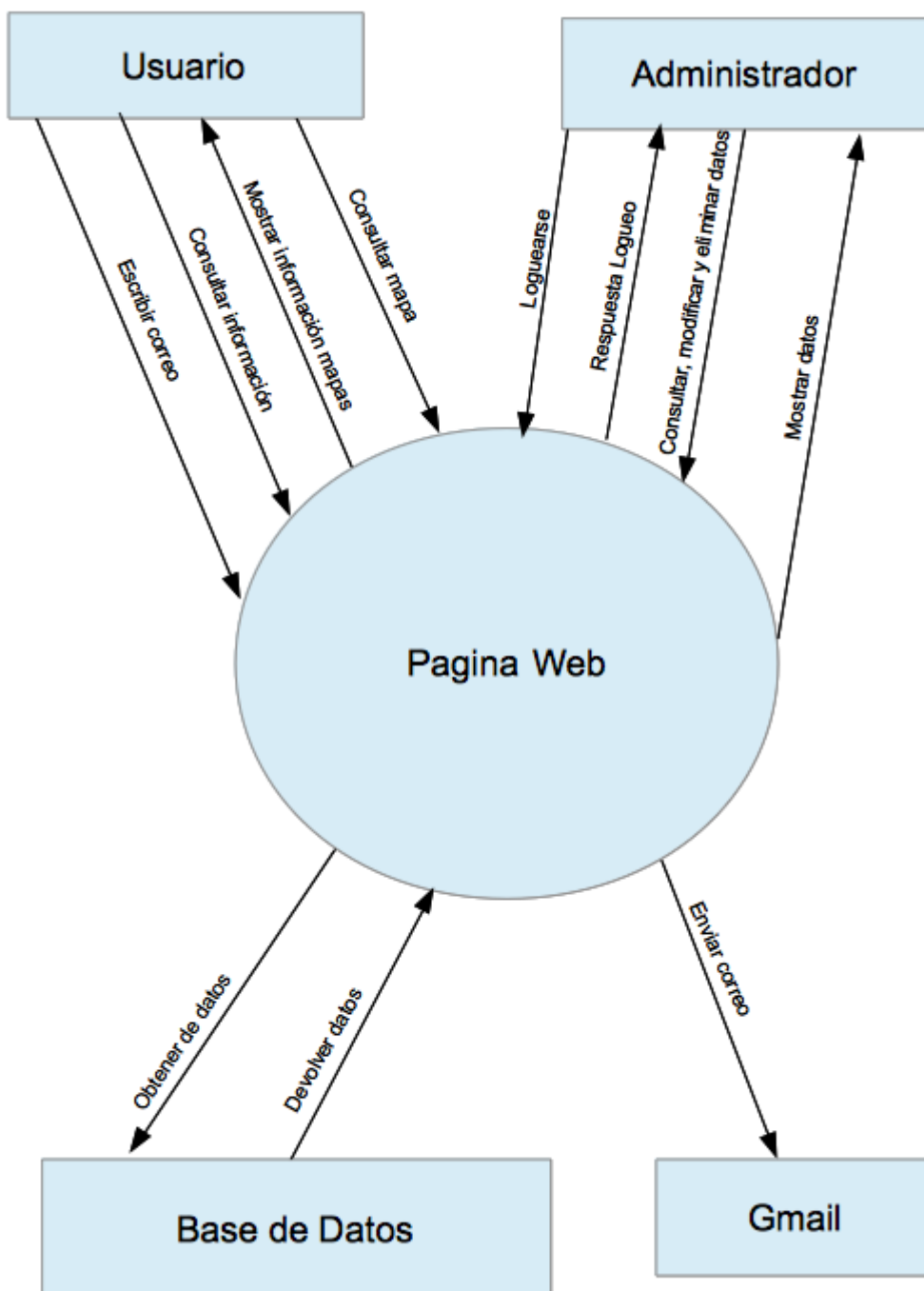


5) Diagrama de clases



6) Esquema Funcional

En esta parte vamos a explicar el funcionamiento, a través de un diagrama de funcional, de nuestra página web y como se relaciona con la base de datos y los correos. Para ello hemos dividido la web en dos. Una parte desde la interfaz de usuario y la otra parte desde la interfaz de administrador.



1. Interfaz de usuario.

Es la parte que corresponde al usuario final, al que va dedicado la web. Se trata de que cualquier usuario se pueda conectar a través de internet a nuestra página web y pueda consultar mediante unas funciones internas con nuestra base de datos y los mapas correspondiente para poder ubicar los edificio o lugares que desea buscar del entorno de la universidad Pablo de Olavide, de Sevilla “UPO”. El usuario no necesita tener que registrarse en nuestra web para poder hacer uso de ella. La únicas funcionalidades que tendrá el usuario será el de consultar el mapa a través de un menú lateral en el cual se mostraran los distintos edificios o lugares que pertenecen al recinto de la UPO. También podrá enviar correos sugerencias a los desarrolladores de la web para informar por si ve algún error o tal.

- **index.html** es la página principal, muestra una aplicación web, donde el usuario realizará la gran mayoría de las acciones. No hace falta estar registrado para acceder a ella, cualquier usuario puede acceder. Aquí es donde el usuario consultará el mapa de la UPO.

La forma en la que esta implementada la aplicación principal consta del framework enyo (explicada en el documento). Los datos del mapa serán tratados por la web desde la carpeta Source donde tenemos varios archivos JavaScript. Para obtener los datos de la BBDD usamos Ajax que funciona por medio de una api que la implementamos en php y se llama api-datos, Esta api nos devuelve un objeto JSON con el cual mostramos los datos de la petición en las listas.

- **informacion.html** en esta página encontraremos información sobre el proyecto web e información de los desarrolladores web que hemos realizado este proyecto. Tampoco necesitamos estar registrados para acceder a esta parte de la aplicación web. Es un simple archivo HTML.
- **sugerencia.php** aquí el usuario podrá enviar correos a los desarrolladores sobre sugerencias o errores pasados por alto. Se trata de una carpeta Sugerencias donde tenemos un archivo php (sugerencias.php), con su respectiva función en JavaScript para validar el formulario, que manda los datos a otro archivo php (enviar.php) el que se encarga de enviar el correo.

2. Interfaz de administrador.

Es la parte correspondiente a las personas que administraran la página web. Estas tarea de administración trata de mantener la base de datos a través de unos controles input y mantener la página web en el correcto funcionamiento. Estos controles estarán controlados para que no se pueda meter cosas incoherentes, con sus respectivas funciones en JavaScript.

En la carpeta Gestión tenemos todos los archivos correspondientes a esta tarea. Por cada entidad tenemos tres archivos (gestion-NombreEntidad-api.php ,gestion-NombreEntidad-script.php, gestion-NombreEntidad.php) tenemos el archivo entrar.php, gestion.php, autenticacion.php y saneo.php.

- **entrar.php** es la encargada de que el usuario administrador meta su usuario y contraseña para poder gestionar la aplicación web. Ésta se encargará de llamar al archivo gestion.php una vez que se haya verificado con una consulta a la base de datos de que ese usuario existe y la contraseña es válida y creará la variable sesión.

Para poder entrar en las siguientes páginas debemos de tener creado la variable sesión de lo contrario nos redirigirá a entrar.php.

- **gestion.php** es la página principal para el administrador. En ella tenemos un menú con las distintas entidades.
- **gestion-NombreEntidad.php** digamos que es la encargada de cargar la página principal de cada entidad.
- **gestion-NombreEntidad-script.php** En esta página tenemos los correspondientes botones e inputs de modificación, la tabla donde mostraremos los datos, validaciones de datos, etc. Para poder modificar o añadir datos tenemos que pasar unas ciertas validaciones las cuales hemos implementado con unas funciones JavaScript en esta misma página. Hemos usado Ajax para el tema de las modificaciones para no tener que volver a cargar la página.

Las consultas para las modificaciones en la base de datos las tenemos implementadas otro archivo distinto llamado gestion-NombreEntidad-api.php.

- **gestion-NombreEntidad-api.php** es un archivo encargado de ejecutar las consultas sql que llegan desde gestion-NombreEntidad.php.
- **saneos.php** es la encargada de que no nos puedan hacer inyección sql.
- **sesion.php** controlamos que los usuarios estén autenticados.
- **salir.php** para destruir la variable sesión.

Para poder acceder a esta parte debemos de estar registrado en la base de datos, ya que esta parte es restringida por un fichero en php.

Todas las páginas las tenemos relacionadas con hojas de estilos.

Anexo: hemos usado muchas funciones las cuales no hemos descrito, pero que la aplicación están comentadas, ya que creemos que están mejor explicado por las personas que la han desarrollado.

Frameworkos adicionales.

Para la presentación principal hemos usado el framework ENYO y para la gestión usamos JQuery.

¿Por qué enyo?

Hemos optado por este framework porque la encapsulación es fácil de implementar y se mantiene de forma ordenada. Por lo tanto el motivo principal es que al trabajar en grupo es más fácil dividirnos el trabajo sin depender de los avances de otros compañeros. Ahora pasamos a explicar el funcionamiento básico del framework.

Enyo es un framework JavaScript, con el cual tenemos la posibilidad del despliegue de la aplicación, tanto para plataformas móviles y plataformas de escritorio (próximamente a TV). El framework nos asegura una amplia compatibilidad con los navegadores más populares del mercado (con algunas excepciones).

Funcionamiento.

En Enyo todo es tratado como objetos. Para crear uno basta con hacer una llamada a “`enyo.kind`”

Un ejemplo de cómo quedaría un objeto creado

```
enyo.kind({  
  name: "holaAPP",  
  components: [  
    {tag: "p", name: "hola", content: "hola mundo"},  
    {kind: "Button", content: "Presiona", ontap: "holaTap"}  
  ],  
  holaTap: function() {  
    this.$.hola.applyStyle("color", "red");  
  }  
});
```

Hemos creado un objeto llamado “holaAPP” el cual tiene elementos hijos que están definidos dentro de “components”. Estos hijos a su vez son objetos los cuales pueden contener métodos y atributos propios.

Para acceder a un objeto solo debemos llamarlo como `this.$.<nombre del objeto>` y seguimos una sintaxis como la de java o c para los métodos.

Cada objeto cuenta con métodos predeterminados, por ejemplo el de cambiar el estilo. Como podemos ver en el ejemplo anterior el objeto “hola” tiene el método “applyStyle” con el que podemos cambiar el estilo del objeto. En el mismo objeto debemos hacer referencia a los eventos que obedece el mismo objeto, este es el caso de “ontap” en el cual se especifica la función a la cual obedece el evento mencionado.

Llamadas a los objetos

Como en Enyo todo es tratado como objetos, vamos a llamar al objeto que antes creamos, este es un ejemplo de cómo se haría.

```
enyo.kind({  
  name: "MyApp",  
  components: [  
    {kind: "holaAPP"},  
    {kind: "holaAPP"}  
  ]  
});
```

Hemos creado otro objeto llamado “MyApp” el cual contiene dos objetos “holaAPP” que son objetos del tipo que creamos antes.

Cada uno de estos objetos cuenta con sus métodos y atributos de forma independiente y también obedecen a métodos de su padre, en este caso “MyAPP”.

Esta es la facilidad que nos brinda Enyo, cada objeto lo podemos almacenar en un archivo distinto y hacer llamadas a sus componentes dentro de otros componentes. Esta es la principal razón por la que seleccionamos el framework para poder trabajar en equipo.

Por ultimo llamamos a nuestro objeto

```
new MyApp().renderInto(document.body);
```

Por su puesto que si queremos introducirnos a fondo en el funcionamiento podemos visitar la página.

<http://enyojs.com>

Con estos principios básicos sobre el framework hemos finalizado con la documentación sobre los framework usados. No nos referimos a JQuery porque es el que vimos en clase.

Servicios adicionales

Para mostrar los mapas hemos usado el servicio de mapbox. Ahora pasamos a explicar el funcionamiento básico del servicio de <https://www.mapbox.com>

Mapbox ofrece una API implementada en JS, también en otros lenguajes, nosotros usamos el de JavaScript.

Primero tenemos que crearnos una cuenta y obtener la apikey, lo más normal es que nuestra APIKEY sea el nombre de usuario seguido del código del mapa que generemos. Por ejemplo usuario.h4n5k2n8.

Para acceder a su API debemos incluir las siguientes metas.

```
<script src='https://api.tiles.mapbox.com/mapbox.js/v1.6.1/mapbox.js'></script>
<link href='https://api.tiles.mapbox.com/mapbox.js/v1.6.1/mapbox.css' rel='stylesheet' />
```

El siguiente paso es crear un “objeto” mapa. Lo hacemos de la siguiente forma.

```
var map = L.mapbox.map('map', usuario.h4n5k2n8);
```

A partir de este momento podemos llamar a los métodos que contiene el objeto, un ejemplo puede ser insertar una ubicación.

```
L.mapbox.markerLayer({
  type: 'Feature',
  geometry: {
    type: 'Point',
    coordinates: [longitud,latitud]
  },
  properties: {
    title: ""
    description: ""
    'marker-size': 'large',
    'marker-color': '#000'
  }
}).addTo(map);
```

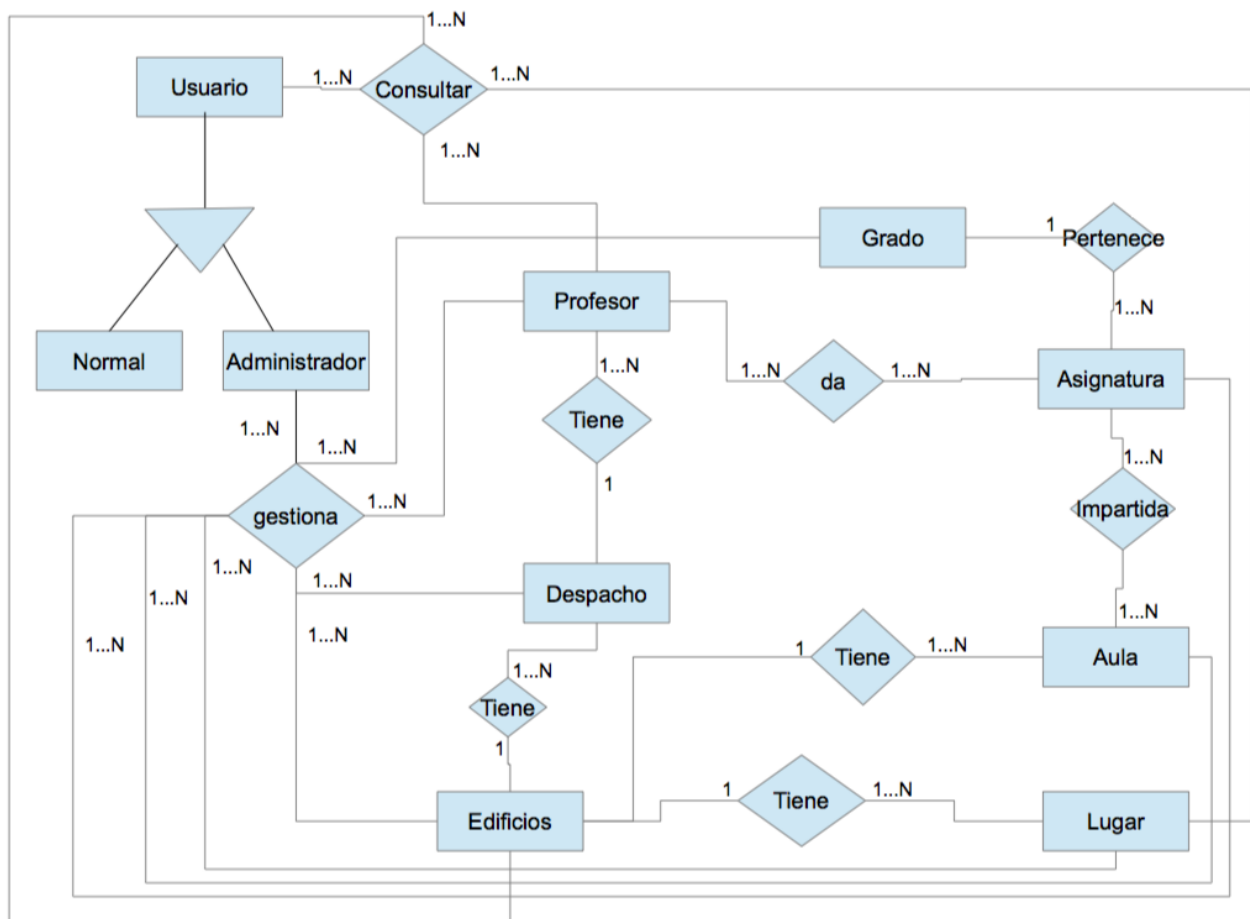
Con esto podemos insertar un ubicación en el mapa, claro que si queremos insertar un conjunto solo debemos meter la instrucción en un bucle.

En conclusión para insertar ubicaciones debemos crear un JSON con el formato indicado y añadirlo al mapa. Para consultar los servicios adicionales podemos visitar

<https://www.mapbox.com/developers/>

Cabe recalcar que para mostrar los datos en la aplicación también usamos Ajax y JSON

7) Diseño entidad-relación



Nota: La relación gestiona debería ser un gestiona para cada entidad y la relación consulta debe tener una relación por cada entidad que se relaciona.

Hemos escogidos estas entidades porque son las necesarias para poder realizar una solución al problema propuesto que era la disposición de todos los puntos, por ello necesitábamos cada una de las entidades, las asociaciones son las siguientes.

- edificios contiene despachos
- edificios contiene aulas
- profesores imparten asignaturas
- asignaturas pertenecen algún grado
- profesor tiene despacho

Por otro lado tenemos a lugares y a usuarios, que, los lugares son gestionados por los usuarios.

8) Manual Usuario

Para la utilización de la herramienta de un usuario corriente solo tiene que acceder a la página web desde un ordenador portátil, móvil o Tablet, pues la interfaz está adaptada para su manejo con pantallas táctiles.

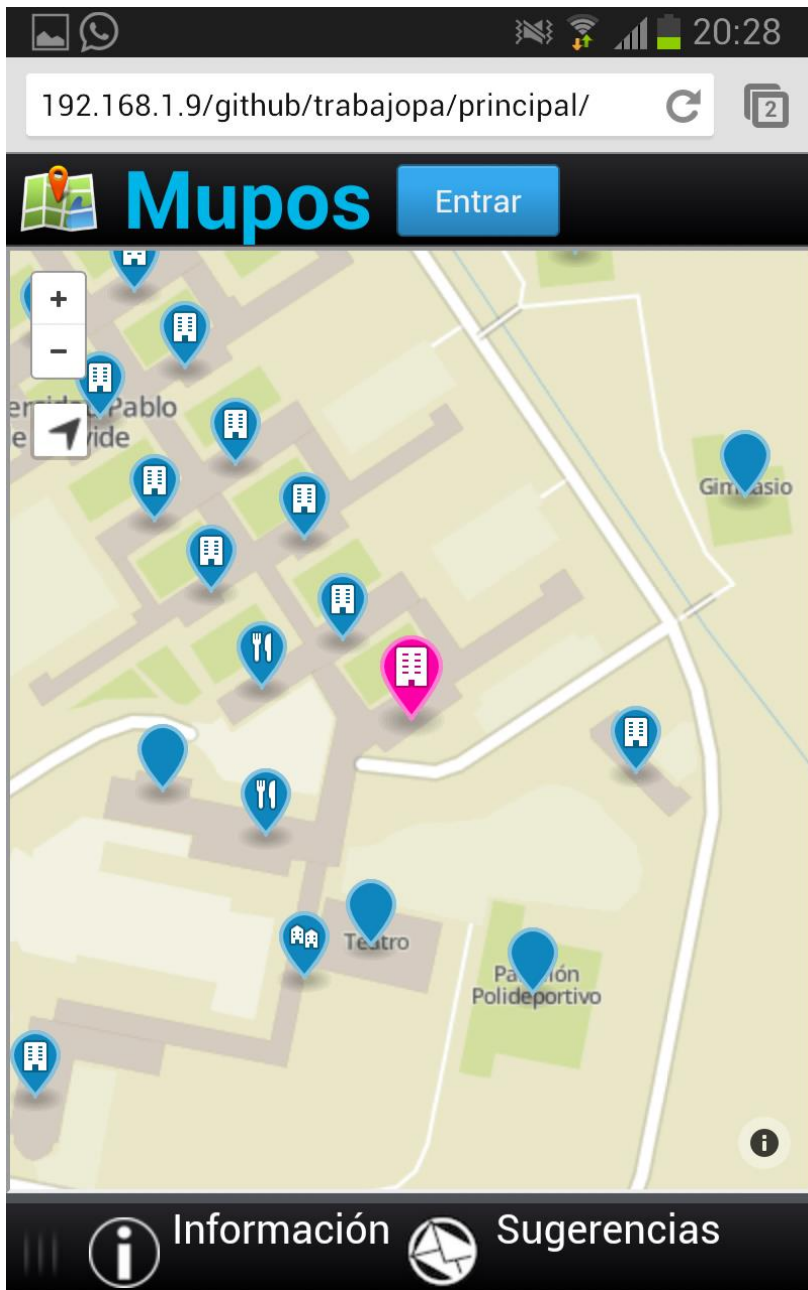
Una vez dentro los primero que se tiene que elegir es que tipo de lugar estas buscando.



Dentro de cada categoría tendremos todos los lugares de esa categoría disponibles para ser vistos en el mapa.

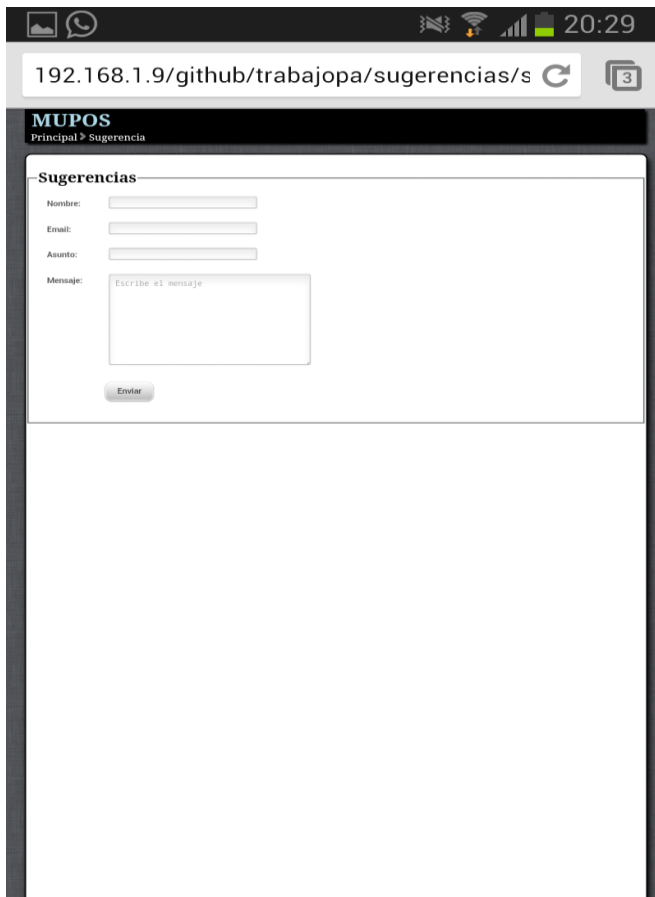


Una vez seleccionado el edificio, es lo que hemos escogido para este manual, o cualquier otra categoría y lugar nos iremos al mapa con su “chincheta” correspondiente marcándonos nuestra búsqueda. En nuestro caso hemos seleccionado edificio 2 “Antonio de Ulloa”.

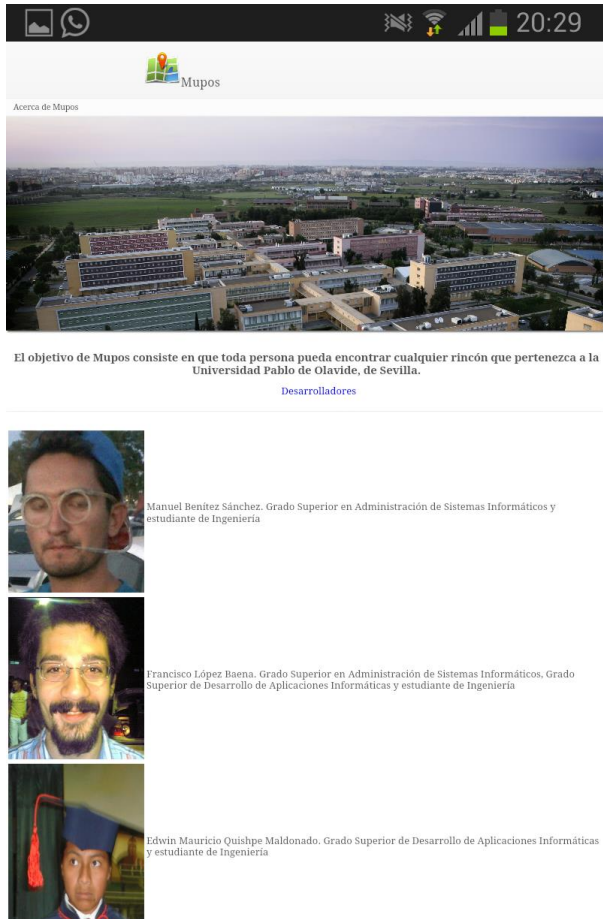


Con lo visto hasta ahora ya podemos buscar lugares que necesitemos encontrar en nuestro campus universitario de la Pablo de Olavide, fácil ¿no?

En la última pantalla podemos ver dos enlaces, con el de sugerencias damos la posibilidad a los usuarios de mandarnos cualquier tipo de duda o sugerencia de mejora de la herramienta, que nos llega directamente a nuestro correo.



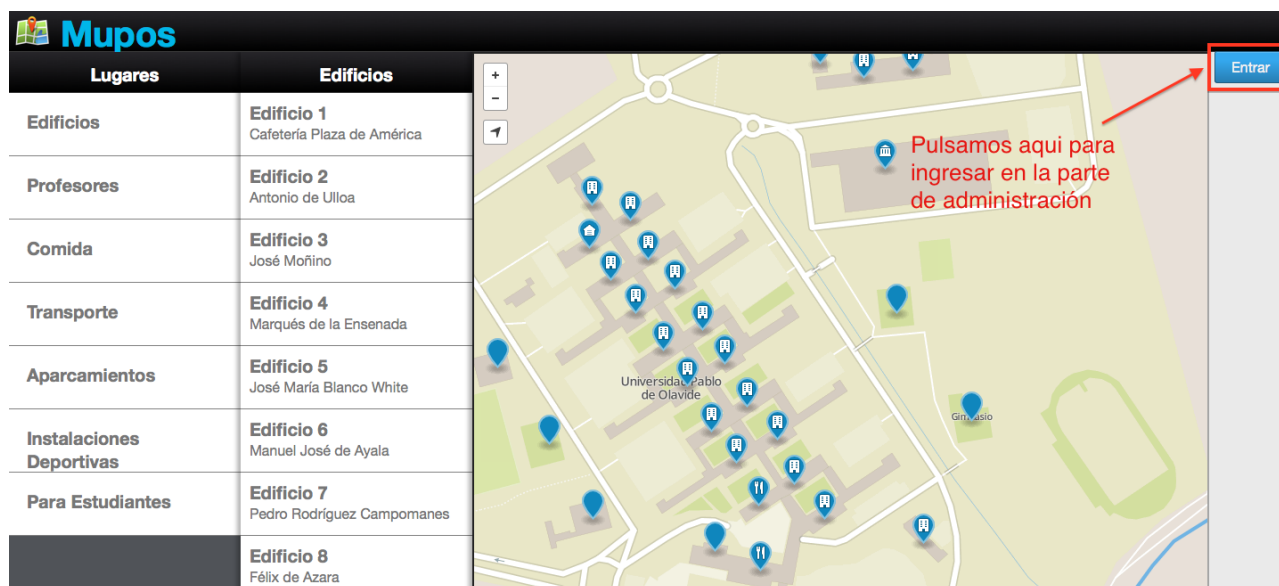
El otro enlace da información sobre nosotros “about us”.



9) Manual de Administrador

El manual de administrador esta explicado en las propias capturas de pantallas.

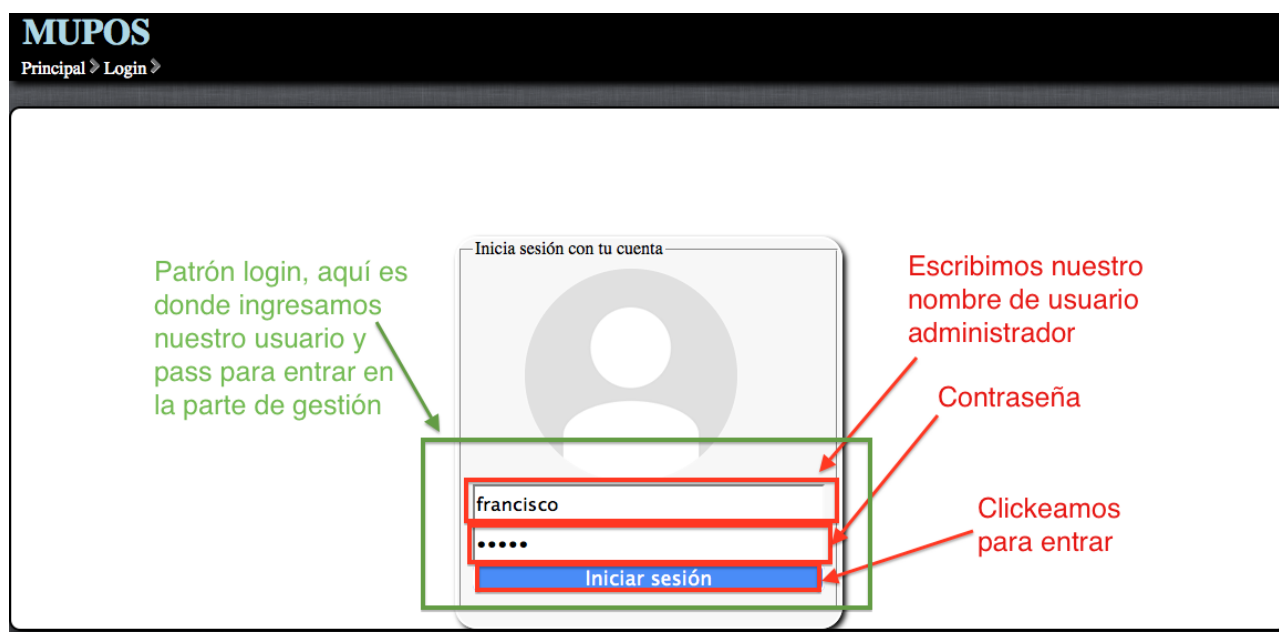
1- Login. Nos logamos en el sistema para poder acceder al área de gestión.



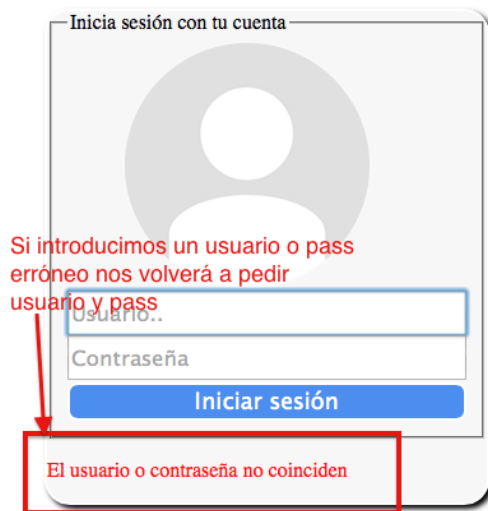
The screenshot shows the Mupos system interface. On the left is a sidebar menu with the following items:

Lugares	Edificios
Edificios	Edificio 1 Cafetería Plaza de América
Profesores	Edificio 2 Antonio de Ulloa
Comida	Edificio 3 José Moñino
Transporte	Edificio 4 Marqués de la Ensenada
Aparcamientos	Edificio 5 José María Blanco White
Instalaciones Deportivas	Edificio 6 Manuel José de Ayala
Para Estudiantes	Edificio 7 Pedro Rodríguez Campomanes
	Edificio 8 Félix de Azara

On the right is a map of the university campus with various buildings marked. A red arrow points to a blue button labeled "Entrar" in the top right corner. A red text box with an arrow pointing to the map says: "Pulsamos aqui para ingresar en la parte de administración".



The screenshot shows the MUPOS login page. The header includes "MUPOS" and "Principal > Login >". The main content area is titled "Inicia sesión con tu cuenta" and features a login form. A green arrow points to the form with the text: "Patrón login, aquí es donde ingresamos nuestro usuario y pass para entrar en la parte de gestión". The form contains two input fields: the first is labeled "francisco" and the second is labeled ".....". A blue button labeled "Iniciar sesión" is at the bottom of the form. Red arrows point to the input fields and the button with the following text: "Escribimos nuestro nombre de usuario administrador", "Contraseña", and "Clickeamos para entrar".



- 2- Una vez logueados en el sistema tendremos un menú para gestionar las diferentes entidades. Dentro de cada entidad podremos crear, eliminar y modificar datos.



- 3- Si picamos en cualquiera de ellos.

ID	Nombre	Primer Apellido	Segundo Apellido	Nombre Asignatura	Numero de Edificio	Numero de Despacho
1	Carlos	Barranco	González	Programación avanzada	13	1.2
2	Manue	Béjar	Domínguez	Fundamentos de Computadores	14	1.25
4	Federico	Divina		Biología	12	1.9
5	Domingo Savio	Rodríguez	Baena	Mantenimiento Bases de datos	11	1.29
6	Sergio	Bermudo	Navarrete	cálculo	3	2.16
8	Francisco	Martínez	Álvarez	Algorítmica 1	9	2.7

Nos muestra los datos que hay en la base de datos sobre profesor

Clickeamos para crear un profesor nuevo



ID	Nombre	Primer Apellido	Segundo Apellido	Nombre Asignatura	Numero de Edificio	Numero de Despacho
1	Carlos	Barranco	González	Programación avanzada	13	1.2
2	Manue	Béjar	Domínguez	Fundamentos de Computadores	14	1.25
4	Federico	Divina		Biología	12	1.9
5	Domingo Savio	Rodríguez	Baena	Mantenimiento Bases de datos	11	1.29
6	Sergio	Bermudo	Navarrete	cálculo	3	2.16
8	Francisco	Martínez	Álvarez	Algorítmica 1	9	2.7

Datos

ID. del Profesor:

Nombre Profesor:

Primer Apellido:

Segundo Apellido:

Nombre Asignatura:

Numero de despacho:

Id del profesor es automático

Nombre del profesor

Apellido 1 (opcional)

Apellido 2 (opcional)

Elegir una asignatura

Elegir número de despacho

Crear el profesor y guardarlo en la base de datos

Crear Profesor

Cancelar

Cancelar la inserción de un nuevo profesor

4- Si picamos en el dato que deseamos modificar o eliminar, nos aparecerá un pequeño menú para poder acceder a esas acciones.

ID	Nombre	Primer Apellido	Segundo Apellido	Nombre Asignatura	Numero de Edificio	Numero de Despacho
1	Carlos	Barranco	González	Programación avanzada	13	1.2
2	Manue	Béjar	Domínguez	Fundamentos de Computadores	14	1.25
4	Federico	Divina		Biología	12	1.9
5	Domingo Savio	Rodríguez	Baena	Mantenimiento Bases de datos	11	1.29
6	Sergio	Bermudo	Navarrete	cálculo	3	2.16
8	Francisco	Martínez	Álvarez	Algorítmica 1	9	2.7

Si clickeamos sobre una fila se marcará en naranja la fila clickeada

Y aparecerán nuevos botones de control para la gestión

Editar, para editar el registro marcado



Eliminar, borrar el registro marcado